

**AMENDMENTS TO THE CLAIMS**

Please amend claim 18 as follows:

1           1. (Original) A method for commonly controlling device drivers, comprising the steps of:  
2           arranging a device independent access hierarchy between an application hierarchy and a  
3           device driver hierarchy and applying a standardized rule of said device independent access hierarchy  
4           to said application hierarchy and said device driver hierarchy; and  
5           allowing said application hierarchy and said device driver hierarchy to access the device  
6           driver hierarchy and said application hierarchy through the standardized rule of said device  
7           independent access hierarchy, respectively.

1           2. (Original) The method as set forth in claim 1, with said step of allowing said application  
2           hierarchy and said device driver hierarchy to access, comprising the steps of:  
3           allowing said application hierarchy to transmit control commands based on a standardized  
4           common format for a corresponding device driver to said device independent access hierarchy, and  
5           allowing said device independent access hierarchy to convert the control commands into other  
6           control commands based on a local format and transmit the converted control commands to said  
7           device driver; and  
8           allowing said device driver to give a response to the converted control commands based on  
9           the local format to said device independent access hierarchy, and allowing the device independent

10 access hierarchy to convert the response from said device driver into a response based on the  
11 standardized common format and transmit the response based on the standardized common format  
12 to said application hierarchy.

1 3. (Original) A method for commonly controlling device drivers, comprising the steps of:  
2 arranging a device independent access hierarchy between an application hierarchy and a  
3 device driver hierarchy;  
4 defining functions available in a corresponding device driver among functions of a function  
5 block in a function table;  
6 when a device is initialized, allowing said device independent access hierarchy to generate  
7 a device handler identifier based on a standardized data format for said device and transmit the  
8 generated device handler identifier to the application hierarchy of a higher order; and  
9 allowing the higher-order application hierarchy to call a predetermined device using the  
10 device handler identifier, and allowing said device independent access hierarchy to identify a  
11 function of the corresponding device driver from the function table using the device handler  
12 identifier and call the function of the corresponding device driver.

1 4. (Original) The method as set forth in claim 3, with said device handler identifier being  
2 represented as DCB handlerId[x1.x2.x3], where x1, x2 or x3 is an unsigned integer, x1 being a value  
3 of the level 1 meaning a device ID, x2 being a value of the level 2 meaning a logical or physical  
4 group number of a corresponding device, x3 being a value of a channel meaning a channel number

5 of a corresponding device or group.

1 5. (Original) The method as set forth in claim 4, with values of x1, x2 and x3 being "0"  
2 corresponding to there being no corresponding level or channel and the value of x1 sequentially  
3 increasing from "1" when the device is initialized.

1 6. (Original) A method for commonly controlling device drivers, comprising the steps of:  
2 arranging a device independent access hierarchy between an application hierarchy and a  
3 device driver hierarchy;

4 when a device initialization is controlled by said application hierarchy, allowing said device  
5 independent access hierarchy to carry out level 1 initialization, level 2 initialization and channel  
6 initialization and generate a device handler identifier based on a standardized data format for a  
7 device;

8 allowing said device independent access hierarchy to dynamically assign a device control  
9 block, containing elements for carrying out a standardized rule, corresponding to said device handler  
10 identifier;

11 allowing said device independent access hierarchy to provide said device handler identifier  
12 to said application hierarchy; and

13 allowing said application hierarchy to call a predetermined device through said device  
14 independent access hierarchy using said device handler identifier.

1           7. (Original) The method as set forth in claim 6, with the elements of said device control  
2 block comprising a pointer of **"\*pControlTable"** for pointing a position of a command control table,  
3 the command control table containing a command identifier having a standardized unique value and  
4 a command function pointer mapped to the command identifier, a pointer of **"\*pDDCB"** for pointing  
5 a position of a device driver control table through which the existence and position of a  
6 corresponding function is identified, and a pointer **"\*pAnchor"** for pointing a next level.

1           8. (Original) The method as set forth in claim 6, with the elements of said device control  
2 block comprising a pointer of **"\*pHandler"** for pointing a position of a given initialization profile  
3 when a device is initialized, a function pointer of **"\*fpInitDevice"** being used when a device is  
4 initialized, a function pointer of **"\*fpOpenChannel"** being used when a channel is open, a function  
5 pointer of **"\*fpCloseChannel"** being used when a channel is closed, a function pointer of **"\*fpRead"**  
6 being used when data of an open channel is read, a function pointer of **"\*fpWrite"** being used when  
7 data of the open channel is written, a function pointer of **"\*fpReset"** being used when a device is  
8 reset, a pointer of **"\*pControlTable"** for pointing a position of a command control table containing  
9 a command identifier having a standardized unique value and a command function pointer mapped  
10 to the command identifier, a pointer of **"\*pDDCB"** for pointing a position of a device driver control  
11 table through which the existence and position of a corresponding function is identified, a pointer  
12 of **"\*pEventTable"** for pointing a position of an event table, and a pointer **"\*pAnchor"** for pointing  
13 a next level.

1           9. (Original) The method as set forth in claim 6, with the level 1 initialization of said device  
2 being made by giving a device identifier value of x1 as a unique value for each device based on a  
3 sequence of the level 1 initialization in the device handler identifier represented as DCB  
4 handlerId[x1.x2.x3] where x1, x2 or x3 is an unsigned integer.

1           10. (Original) The method as set forth in claim 9, with the level 2 initialization of the device  
2 being made by referring to the number of logical or physical groups, assigning anchors, and giving  
3 a group value of x2 as a unique value for each anchor in the device handler identifier represented as  
4 DCB handlerId[x1.x2.x3] where x1, x2 or x3 is an unsigned integer.

1           11. (Original) The method as set forth in claim 10, with the level 3 initialization of the  
2 device being made by giving a channel value of x3 for each of channels belonging to the device and  
3 groups within the device on the basis of an open channel sequence in the device handler identifier  
4 represented as DCB handlerId[x1.x2.x3] where x1, x2 or x3 is an unsigned integer.

1           12. (Original) A method, comprising:  
2 requesting loss of signal state information based on a standardized common format by an  
3 application to a device independent access hierarchy;  
4 converting the request from said application into a first device local format and requesting  
5 a first device driver to provide the loss of signal state information to said device independent access  
6 hierarchy;

7           responding to the request for loss of signal state information based on the first device local  
8   format;

9           responding to said application by said device independent access hierarchy for loss of signal  
10   state information based on the standardized common format.

1           13. (Original) The method of claim 12, with said step of converting the request from said  
2   application further comprising of converting the request into a second device local format and  
3   requesting a second device driver to provide the loss of signal state information to said device  
4   independent access hierarchy based on the second device local format when a first device is  
5   converted to a second device and said first device driver is changed to said second device driver.

1           14. (Original) The method of claim 13, further comprising of converting control commands  
2   based on the standardized common format to control commands provided to the device drivers  
3   accommodating a change of said application to a second application without changing the control  
4   commands provided to the device drivers.

1           15. (Original) The method of claim 14, further comprised of providing a mutual interface  
2   between said application and said first and second device drivers by the device independent access  
3   hierarchy reading material from a device driver control block and accessing the first and second  
4   device drivers using predetermined functions.

1           16. (Original) The method of claim 15, further comprising of said device independent access  
2 hierarchy using device handler identifiers based on the standardized data format, said device handler  
3 identifiers corresponding to respective devices.

1           17. (Original) The method of claim 16, further comprising:  
2 providing the device handler identifiers to said application from said device independent  
3 access hierarchy during an initialization of the corresponding device; and  
4 storing, by said application, the device handler identifiers and calling a corresponding device  
5 using a corresponding device handler identifier.

1           18. (Currently Once Amended) The method of claim 17, further comprising of said device  
2 independent access hierarchy determining according to said device handler identifier whether a  
3 certain device driver should be called and calling the certain driver handler device driver according  
4 to the determination.

1           19. (Original) The method of claim 18, with the device independent access hierarchy using  
2 certain pointers and function pointers in performing the standardized common format in the device  
3 independent access hierarchy.

1           20. (Original) The method of claim 19, further comprised of when said application is calling  
2 a function of a function block to be used, said device independent access hierarchy identifies the

3     existence of a corresponding function from a function table and uses a device handler identifier to  
4     inform the initialization of the device driver accommodating said application to access a device  
5     driver using said device handler identifier.

1           21. (Original) The method of claim 20, further comprised of not varying the device handler  
2     identifier value for the device when said first device driver is changed to said second device driver.

1           22. (Original) The method of claim 21, further comprising of varying the addresses of the  
2     pointers under the control of said device independent access hierarchy when said first device driver  
3     is changed to said second device driver.